

MUSICAL SCORE DISPLAYING METHOD

Patent Number: JP10198352

Publication date: 1998-07-31

Inventor(s): NISHIKAWA MASASHI; WATASE TAKAO; FUNADA KOICHI

Applicant(s): ROLAND CORP

Requested Patent: JP10198352

Application Number: JP19970017313 19970114

Priority Number(s):

IPC Classification: G10G3/04; G09G5/14; G10H1/00

EC Classification:

Equivalents:

Abstract

PROBLEM TO BE SOLVED: To display a musical score such that a user easily recognizes its contents without defects, by calculating number of small sections to be displayed on the basis of the number of the sections for each line of the musical score and the number of lines to be displayed in display area on a display screen, and by displaying the small sections.

SOLUTION: Various switches 108, 110, 112 operable by a mouse are displayed on a window. Part other than display area such as the switches 108, 110, 112, clefs, or margin is set as a musical note displaying region 100. Because a resize box 106 is displayed, a user puts the mouse on the box 106 and moves the mouse clicking to change a shape on the window. A musical score is displayed corresponding to a specified number of small sections and that of lines. This process requires variables such as size of musical note displaying region and small sections, the number of small sections per line, the number of lines, or front size.

Data supplied from the esp@cenet database - I2

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-198352

(43)公開日 平成10年(1998)7月31日

(51)Int.Cl.⁶
G 1 0 G 3/04
G 0 9 G 5/14
G 1 0 H 1/00

識別記号

F I
G 1 0 G 3/04
G 0 9 G 5/14
G 1 0 H 1/00

A
Z

審査請求 未請求 請求項の数6 FD (全25頁)

(21)出願番号 特願平9-17313

(22)出願日 平成9年(1997)1月14日

(71)出願人 000116068
ローランド株式会社
大阪府大阪市北区堂島浜1丁目4番16号

(72)発明者 西川 雅士
大阪府大阪市北区堂島浜1丁目4番16号
ローランド株式会社内

(72)発明者 渡瀬 孝雄
大阪府大阪市北区堂島浜1丁目4番16号
ローランド株式会社内

(72)発明者 船田 浩一
大阪府大阪市北区堂島浜1丁目4番16号
ローランド株式会社内

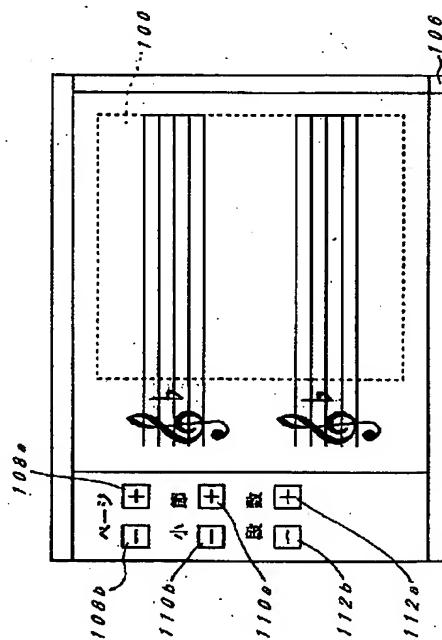
(74)代理人 弁理士 上島 淳一

(54)【発明の名称】 楽譜表示方法

(57)【要約】

【課題】画面の表示領域がどのような形状やサイズであっても楽譜が欠けて表示されることはなく、ユーザーが視認し易い形態で楽譜を表示する。

【解決手段】コンピュータにより制御されるディスプレイの画面上の表示領域に楽譜を表示するための楽譜表示方法において、楽譜の1段当たりの小節数を示す情報とディスプレイの画面上の表示領域に表示する段数を示す情報とに基づいて、上記表示領域に表示すべき小節数を算出し、上記算出した小節数の小節を上記表示領域に表示するようにした。



【特許請求の範囲】

【請求項1】 コンピュータにより制御されるディスプレイの画面上の表示領域に楽譜を表示するための楽譜表示方法において、楽譜の1段当たりの小節数を示す情報とディスプレイの画面上の表示領域に表示する段数を示す情報とに基づいて、前記表示領域に表示すべき小節数を算出し、前記算出した小節数の小節を前記表示領域に表示することを特徴とする楽譜表示方法。

【請求項2】 請求項1記載の楽譜表示方法において、前記小節の高さを示す情報と前記表示領域の高さを示す情報とに基づいて、前記表示領域に表示可能な段数を算出することを特徴とする楽譜表示方法。

【請求項3】 請求項2記載の楽譜表示方法において、前記小節の縦横比を示す情報と前記小節の高さを示す情報と前記表示領域の幅を示す情報とに基づいて、前記表示領域における1段当たりに表示可能な小節数を算出することを特徴とする楽譜表示方法。

【請求項4】 請求項3記載の楽譜表示方法において、前記小節の縦横比を示す情報と前記表示領域の幅を示す情報とに基づいて、前記表示領域に表示可能な小節の高さを算出することを特徴とする楽譜表示方法。

【請求項5】 コンピュータにより制御されるディスプレイの画面上の表示領域に楽譜を表示するための楽譜表示方法において、段の縦横比を示す情報とディスプレイの画面上の表示領域の幅を示す情報とに基づいて、前記表示領域に表示可能な前記段の高さを算出し、前記算出した段の高さと前記表示領域の高さとに基づいて、前記表示領域に表示可能な段数を算出することを特徴とする楽譜表示方法。

【請求項6】 コンピュータにより制御されるディスプレイの画面上の表示領域に楽譜を表示するための楽譜表示方法において、段の高さが指定されたとき、

ディスプレイの画面上の表示領域の幅を示す情報と前記指定された段の高さを示す情報とに基づいて、前記段の縦横比を可変制御するとともに、

前記表示領域の幅が指定されたとき、前記段の縦横比を示す情報と前記指定された表示領域の幅を示す情報とに基づいて、前記表示領域に表示可能な段の高さを算出し、

前記算出した段の高さと前記表示領域の高さとに基づいて、前記表示領域に表示可能な段数を算出することを特徴とする楽譜表示方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、楽譜表示方法に関し、さらに詳細には、パーソナルコンピュータなどの汎用のコンピュータに備えられたLCDやCRTなどのデ

ィスプレイの画面上に楽譜を表示するための楽譜表示方法に関する。

【0002】

【従来の技術】従来より、パーソナルコンピュータなどの汎用のコンピュータに備えられたLCDやCRTなどのディスプレイの画面上に楽譜を表示し、表示した楽譜を用紙に印刷することができるようになした楽譜表示方法が知られているが、こうした楽譜表示方法においては、画面の表示領域に関わらず、楽譜を印刷する用紙のサイズに合わせて、小節数や段数を設定するようになされている。

【0003】このため、こうした従来の楽譜表示方法にあっては、画面の表示領域と用紙の形状とが異なると、画面の表示領域には、図1に示すように用紙の一部分が表示されるか、図2に示すように用紙全体が縮小されて表示されるかするようになるものであった。

【0004】即ち、従来の楽譜表示方法のように、楽譜を印刷する用紙のサイズに合わせて、小節数や段数を設定する方法では、画面の表示領域の形状やサイズによって、画面の表示領域においては楽譜の一部が欠けたり、あるいは縮小されて表示されたりすることになり、ユーザーが視認し難いという問題点があった。

【0005】

【発明が解決しようとする課題】そこで、本発明は、従来の技術の有する上記したような種々の問題点に鑑みてなされたものであり、その目的とするところは、画面の表示領域がどのような形状やサイズであっても楽譜が欠けて表示されることなく、ユーザーが視認し易い形態で楽譜を表示することができるようになした楽譜表示方法を提供することにある。

【0006】

【課題を解決するための手段】上記目的を達成するために、本発明のうち請求項1に記載の発明は、コンピュータにより制御されるディスプレイの画面上の表示領域に楽譜を表示するための楽譜表示方法において、楽譜の1段当たりの小節数を示す情報とディスプレイの画面上の表示領域に表示する段数を示す情報とに基づいて、上記表示領域に表示すべき小節数を算出し、上記算出した小節数の小節を上記表示領域に表示するようにしたものである。

【0007】ここで、請求項1に記載の発明において、例えば、本発明のうち請求項2に記載の発明のように、上記小節の高さを示す情報と上記表示領域の高さを示す情報とに基づいて、上記表示領域に表示可能な段数を算出するようにしてもよい。

【0008】また、請求項2に記載の発明において、例えば、本発明のうち請求項3に記載の発明のように、上記小節の縦横比を示す情報と上記小節の高さを示す情報と上記表示領域の幅を示す情報とに基づいて、上記表示領域における1段当たりに表示可能な小節数を算出する

ようにもよい。

【0009】さらに、請求項2に記載の発明において、例えば、本発明のうち請求項4に記載の発明のように、上記小節の縦横比を示す情報と上記表示領域の幅を示す情報とに基づいて、上記表示領域に表示可能な小節の高さを算出するようにしてもよい。

【0010】また、本発明のうち請求項5に記載の発明は、コンピュータにより制御されるディスプレイの画面上の表示領域に楽譜を表示するための楽譜表示方法において、段の縦横比を示す情報とディスプレイの画面上の表示領域の幅を示す情報とに基づいて、上記表示領域に表示可能な上記段の高さを算出し、上記算出した段の高さと上記表示領域の高さとに基づいて、上記表示領域に表示可能な段数を算出するようにしたものである。

【0011】また、本発明のうち請求項6に記載の発明は、コンピュータにより制御されるディスプレイの画面上の表示領域に楽譜を表示するための楽譜表示方法において、段の高さが指定されたとき、ディスプレイの画面上の表示領域の幅を示す情報と上記指定された段の高さを示す情報とに基づいて、上記段の縦横比を可変制御するとともに、上記表示領域の幅が指定されたとき、上記段の縦横比を示す情報と上記指定された表示領域の幅を示す情報とに基づいて、上記表示領域に表示可能な段の高さを算出し、上記算出した段の高さと上記表示領域の高さとに基づいて、上記表示領域に表示可能な段数を算出するようにしたものである。

【0012】

【発明の実施の形態】以下、図面に基づいて、本発明による楽譜表示方法の実施の形態の一例を詳細に説明するものとする。なお、以下の説明においては、各構成の中で同一あるいは相当する構成に関しては、それぞれ共通の符号を付して示すものとする。

【0013】図3は、本発明による楽譜表示方法を実施するためのパーソナルコンピュータなどの汎用のコンピュータの全体構成を示すブロック構成図であり、このコンピュータは、CPU10と、RAM12と、外部記憶装置14と、キーボード16と、マウス18と、ディスプレイ20とを備えている。

【0014】ここで、CPU10は、コンピュータ全体の制御をOS(Operating System)の処理に従って行うものである。OSは、外部記憶装置14やキーボード16ならびにマウス18の入出力の処理、本発明による楽譜表示方法の要旨たるディスプレイに楽譜を表示させるための処理を実行する楽譜表示プログラムの実行制御ならびにディスプレイ20の画面の表示制御を行うとともに、ディスプレイ20の画面上にウィンドウを表示し、さらに、ウィンドウが画面をはみ出した場合のクリッピング処理なども行う。

【0015】また、OSは、ディスプレイ20の画面上にマウスカーソルを表示して、ユーザーのマウス18の

操作に応じた入出力の処理やウィンドウの形状変更の処理などを行う。

【0016】ところで、楽譜表示プログラムは、ユーザーの起動指示に応じてOSから起動され、OSと情報を交換し合うことにより、演算に必要な情報を取得したり、楽譜をウィンドウに表示したりする。

【0017】RAM12は、外部記憶装置14から読み込まれたOSや楽譜表示プログラムならびに楽譜データを格納するとともに、楽譜表示プログラムなどが用いる各種変数を格納するものである。

【0018】外部記憶装置14は、OS、楽譜表示プログラムならびに楽譜データを記憶するものである。

【0019】キーボード16は、数値や文字の入力、ディスプレイ20の画面の入力要求への応答などを行うものである。

【0020】マウス18は、ディスプレイ20の画面上のマウスカーソルの移動制御、マウスカーソルの表示位置での選択および選択取り消しなどを行うものである。

【0021】ディスプレイ20は画面を備え、当該画面にOSにより制御されたウィンドウを表示するとともに、楽譜表示プログラムに従って楽譜などを表示する。

【0022】ここで、以下に詳述する本発明による楽譜表示方法の理解を容易にするために、当該楽譜表示方法による楽譜表示の処理の原理を、図4ならびに図5を参照しながら説明するものとする。

【0023】図4において、音符表示領域100は、5線の音部記号や調性記号などの表示部分を除いた長方形の領域であり、横方向にHwドットとし、縦方向にVwドットとする。また、1段の縦、即ち、小節の縦はVmドットとし、小節の横はHmドットとする。

【0024】そして、1段の中は、5線の上下に2本の加線102を加えた9本の線によって、10個の線間104に等分に区切られる。なお、各線間104は、3度の音高差を備える。

【0025】また、各小節内では、4分音符を24等分したクロックという単位を横方向の表示単位とする。

【0026】従って、Vmドットの縦の高さの中に5線をひき、Hmドットの小節の横幅で小節線をひくことによって、小節が表示されることになる。

【0027】また、小節内には、音符だけではなく、休符や強弱記号なども表示されるが、こうした音符、休符あるいは強弱記号など(以下、音符、休符あるいは強弱記号などを総称して、「記号」と称する。)は、図5に示すように、小節:クロック 音高 記号種というフォーマットで記憶されているものとする。なお、記号種とは、音符、休符、強弱記号などの各記号の種類を示すものである。

【0028】ここで、各記号の大きさは、fドットで示されるフォントサイズにより指定される。ここで、fは

正の整数であり、線間に対応した値が指定される。ただし、「 $f = 0$ 」の場合は、「 $f = 1$ 」に相当するフォントサイズで表示するものとする。

【0029】従って、各記号においては、「小節：クロック」から小節内の横方向の表示位置が指定され、音高から縦方向の表示位置が指定され、その大きさは線間に対応したフォントサイズで指定されることになる。

【0030】以下においては、こうした原理に基づく本発明による楽譜表示方法を、発明の実施の形態のそれぞれに関してより詳細に説明する。(1) 第1の実施の形態：指定された小節数と段数とに対応して楽譜表示を行う場合の処理

図6には、第1の実施の形態における楽譜表示プログラムによって表示されるウィンドウが示されている。このウィンドウ内には、マウス18によって操作可能な各種スイッチが表示されている。ユーザーはマウス18によってこれらのスイッチを操作することにより、所望の動作を指定できる。

【0031】また、これらのスイッチや音部記号、余白などの表示領域を除いた部分が音符表示領域100として設定されている。従って、音符表示領域100は、ウィンドウの形状変化に応じて、その形状が変化する。

【0032】さらに、リサイズボックス106が表示されており、ユーザーはこのリサイズボックス106にマウス18を合わせて、クリックしながら移動(ドラッグ)させることにより、このウィンドウの形状変更を指定できる。なお、ウィンドウの形状変化の情報はOSの処理によって検出され、楽譜表示プログラムに与えられる。

【0033】ここで、図6のウィンドウに示されるスイッチを説明すると、ページ+スイッチ108aは表示された楽譜のページを進めるスイッチであり、ページ-スイッチ108bは表示された楽譜のページを戻すスイッチであり、小節+スイッチ110aは1段当たりの表示小節数を増加させるスイッチであり、小節-スイッチ110bは1段当たりの表示小節数を減少させるスイッチであり、段数+スイッチ112aは表示段数を増加させるスイッチであり、段数-スイッチ112bは表示段数を減少させるスイッチである。

【0034】なお、ページ+スイッチ108aとページ-スイッチ108bとを総称してページスイッチ(ページSW)と称し、小節+スイッチ110aと小節-スイッチ110bとを総称して小節スイッチ(小節SW)と称し、段数+スイッチ112aと段数-スイッチ112bとを総称して段数スイッチ(段数SW)と称する。そして、指定された小節数と段数とに対応して楽譜表示を行う場合の処理においては、以下に示す変数が使用される。

【0035】即ち、

・音符表示領域サイズ 横方向 Hwドット、縦方

向 Vwドット

・小節サイズ

横方向 Hmドット、縦方

向 Vmドット

・1段当たりの小節数

m

・段数

n

・1ページ当たりの小節数

D

・フォントサイズ

f (fは正の整数)

・表示開始小節

St

・最終小節

End

の変数が使用される。

【0036】図7には、ユーザーの起動指示によりOSから起動される楽譜表示プログラムのルーチンのフローチャートが示されている。

【0037】即ち、ユーザーの起動指示によって、楽譜表示プログラムのルーチンがOSから起動されると、まず、各種変数の初期設定を行い、OSより供給されたウィンドウサイズに基づいて、音符表示領域サイズHw、Vwを設定する(ステップS702)。

【0038】それから、外部記憶装置14より、楽譜データに含まれる拍子、音部記号、調性記号などの初期化データを読み込む(ステップS704)。

【0039】次に、変数の初期値として、m=4、n=2、St=1を設定し、Hm=Hw/m、Vm=Vw/n、f=Int(Vm/10)を算出する。ここで、fは、四捨五入された整数值である(ステップS706)。

【0040】上記した各種の初期状態における設定を終了すると、この楽譜表示プログラムのルーチンのサブルーチンたる表示ルーチン(図8参照)を実行し、図8を参照しながら後述するように、算出された小節サイズに基づいて、5線と小節を表示し、音部記号、調性記号などを表示するとともに、外部記憶装置14より小節毎に楽譜データを読み込み、小節サイズ、フォントサイズに基づいて音符などの楽音記号を表示する(ステップS708)。

【0041】この表示ルーチンを終了すると、ページSWがオン(ON)されたか否かを判断し(ステップS710)、ページSWがオンされたと判断された場合には、この楽譜表示プログラムのルーチンのサブルーチンたるページSW処理ルーチン(図9)を実行する(ステップS712)。そして、ページSW処理ルーチンを実行した後は、ステップS708の表示ルーチンへ戻って処理を繰り返す。

【0042】なお、ページSW処理ルーチンにおいては、図9を参照しながら後述するように、押されたページSWに応じて、音符表示領域全体を1ページとして、1ページ分の表示小節を進めたり、戻したりする処理を行う。

【0043】一方、ステップS710において、ページSWはオンされていないと判断された場合には、小節S

Wがオンされたか否かを判断する(ステップS714)。

【0044】そして、ステップS714において、小節SWがオンされたと判断された場合には、この楽譜表示プログラムのルーチンのサブルーチンたる小節SW処理ルーチン(図10)を実行する(ステップS716)。そして、小節SW処理ルーチンを実行した後は、ステップS708の表示ルーチンへ戻って処理を繰り返す。なお、小節SW処理ルーチンにおいては、図10を参照しながら後述するように、押された小節SWに応じて、1段当たりの小節数を増減する。

【0045】一方、ステップS714において、小節SWはオンされていないと判断された場合には、段数SWがオンされたか否かを判断する(ステップS718)。

【0046】そして、ステップS718において、段数SWがオンされたと判断された場合には、この楽譜表示プログラムのルーチンのサブルーチンたる段数SW処理ルーチン(図11)を実行する(ステップS720)。そして、段数SW処理ルーチンを実行した後は、ステップS708の表示ルーチンへ戻って処理を繰り返す。なお、段数SW処理ルーチンにおいては、図11を参照しながら後述するように、押された段数SWに応じて、1ページ内の表示段数を増減する。

【0047】一方、ステップS718において、段数SWはオンされていないと判断された場合には、ウィンドウ(Window)の変化があったか否かを判断する(ステップS722)。即ち、OSにウィンドウサイズが変更されたか否かを問い合わせる。

【0048】そして、ステップS722において、ウィンドウサイズが変更されたと判断された場合には、この楽譜表示プログラムのルーチンのサブルーチンたるWindow変更処理ルーチン(図12)を実行する(ステップS724)。そして、Window変更処理ルーチンを実行した後は、ステップS708の表示ルーチンへ戻って処理を繰り返す。

【0049】なお、Window変更処理ルーチンにおいては、図12を参照しながら後述するように、OSから変更されたウィンドウサイズを受け取って新たな音符表示領域を設定し、新たな音符表示領域に収まるように変数を再計算する。

【0050】なお、ページSW、小節SWあるいは段数SWのいずれもオンではなく、ウィンドウサイズも変更されていない場合には、ステップS710へ戻ってループして待機する。

【0051】次に、図8に示すフローチャート参照しながら、表示ルーチンについて説明する。

【0052】この表示ルーチンにおいては、まず、段数のカウンタiを0にクリアし(ステップS802)、小節数のカウンタjを0にクリアする(ステップS804)。次に、ウィンドウ内の各種スイッチを表示すると

ともに、縦が $i \times Vm$ ドットから $(1+i) \times Vm$ ドットまでの高さVmの表示領域に5線を表示し、音部記号、調性記号などを音符表示領域100外の5線上の所定の位置に表示する(ステップS806)。

【0053】それから、音符表示領域100の5線上にHmドットの間隔で小節線を表示する(ステップS808)。

【0054】次に、外部記憶装置14より $St + (i \times m) + j$ 小節の楽譜データを読み出し、クロック、音高、記号種、フォントサイズfに従って、音符などの記号を表示する(ステップS810)。

【0055】そして、jを1だけインクリメントし(ステップS812)、その後に、 $St + (i \times m) + j > End$ であるか否かを判断する(ステップS814)。

【0056】ステップS814において、 $St + (i \times m) + j > End$ であると判断されて最終小節を超えていた場合には、終止線を小節の終わりに表示するとともにその後の5線を消去して(ステップS816)、楽譜表示プログラムのルーチン(図7)ヘリターンする。

【0057】一方、 $St + (i \times m) + j > End$ でないと判断された場合には、 $j \geq m$ であるか否かを判断する(ステップS818)。

【0058】ステップS818において、 $j \geq m$ であると判断されて1段落分の小節の表示を終えた場合には、iを1だけインクリメントし(ステップS820)、それから $i \geq n$ であるか否かを判断する(ステップS822)。

【0059】そして、ステップS822において、 $i \geq n$ であると判断されて全ての段の表示を終えた場合には、楽譜表示プログラムのルーチン(図7)ヘリターンする。一方、ステップS822において、 $i \geq n$ でないと判断されて全ての段の表示を終えていない場合には、ステップS804へ戻って処理を繰り返す。

【0060】また、ステップS818において、 $j \geq m$ でないと判断されて1段落分の小節の表示を終えていない場合には、ステップS810へ戻って処理を繰り返す。

【0061】上記したように、表示ルーチン(図8)においては、小節サイズ(Hm, Vm)に基づいて、現在の音符表示領域、即ち、1ページに、1段m小節でn段の小節を表示し、ここに音符などの記号を表示することによって、1ページ分の楽譜を表示する処理を行っている。

【0062】次に、図9に示すフローチャート参照しながら、ページSW処理ルーチンについて説明する。

【0063】このページSW処理ルーチンにおいては、まず、1ページ当たりの小節数Dを、 $m \times n$ により算出する(ステップS902)。

【0064】次に、オンされたページSWはページ+スイッチ108aであるか否かを判断する(ステップS9

04)。

【0065】ステップS904において、オンされたページSWはページ+スイッチ108aでない、即ち、オンされたページSWはページースイッチ108bであると判断されてページを戻す場合には、表示開始小節Stが1であるか否かを判断する(ステップS906)。

【0066】ステップS906において、表示開始小節Stは1であると判断されて最初の小節が表示されている場合には、楽譜表示プログラムのルーチン(図7)へリターンする。

【0067】一方、表示開始小節Stは1でないと判断された場合には、St-D<1であるか否かを判断する(ステップS908)。

【0068】ステップS908において、St-D<1でないと判断された場合には、St=St-Dとして、1ページ分の小節数だけ表示開始Stを減らし、楽譜表示プログラムのルーチン(図7)へリターンする(ステップS910)。

【0069】一方、ステップS908において、St-D<1であると判断された場合には、表示開始小節Stを1に設定し、楽譜表示プログラムのルーチン(図7)へリターンする(ステップS912)。

【0070】また、ステップS904において、オンされたページSWはページ+スイッチ108aであると判断されてページを進める場合には、St+D≥Endであるか否かを判断する(ステップS914)。

【0071】ステップS914において、St+D≥Endであると判断されて最終小節が既に表示されている場合には、楽譜表示プログラムのルーチン(図7)へリターンする。

【0072】一方、ステップS914において、St+D≥Endでないと判断された場合には、St=St+Dとして、1ページ分の小節数だけ表示開始Stを増やし、楽譜表示プログラムのルーチン(図7)へリターンする(ステップS916)。即ち、このページSW処理ルーチンにおいては、ページSWによりページの変更が指示されたので、それに応じて表示開始小節Stを1ページ分の小節数Dにより加減している。これによって、新たな表示開始小節Stが設定され、実際の表示は表示ルーチン(図8)で更新される。

【0073】次に、図10に示すフローチャート参照しながら、小節SW処理ルーチンについて説明する。

【0074】この小節SW処理ルーチンにおいては、まず、オンされた小節SWは小節+スイッチ110aであるか否かを判断する(ステップS1002)。

【0075】ステップS1002において、オンされた小節SWは小節+スイッチ110aでない、即ち、オンされた小節SWは小節ースイッチ110bであると判断されて1段当たりの小節数を減少させる場合には、現在の小節数mが1以下(m≤1)であるか否かを判断する

(ステップS1004)。

【0076】ステップS1004において、現在の小節数mが1以下(m≤1)であると判断された場合には、小節数をそれ以上減少させないので、楽譜表示プログラムのルーチン(図7)へリターンする。

【0077】一方、ステップS1004において、現在の小節数mが1以下(m≤1)でないと判断された場合には、mを1だけデクリメントし(ステップS1006)、ステップS1010へ進む。

【0078】また、ステップS1002において、オンされた小節SWは小節+スイッチ110aであると判断されて1段当たりの小節数を増加させる場合には、mを1だけインクリメントし(ステップS1008)、ステップS1010へ進む。

【0079】ステップS1010においては、小節サイズの横方向Hm=Hw/mの演算を行い小節サイズの横方向Hmを算出し、楽譜表示プログラムのルーチン(図7)へリターンする。

【0080】即ち、この小節SW処理ルーチンにおいては、小節SWにより1段当たりの小節数の変更が指示されたので、それに応じて新たな小節サイズの横方向Hmを算出している。これによって、新たな小節サイズの横方向Hmが設定され、実際の表示は表示ルーチン(図8)で更新される。

【0081】次に、図11に示すフローチャート参照しながら、段数SW処理ルーチンについて説明する。

【0082】この段数SW処理ルーチンにおいては、まず、オンされた段数SWは段数+スイッチ112aであるか否かを判断する(ステップS1102)。

【0083】ステップS1102において、オンされた段数SWは段数+スイッチ112aでない、即ち、オンされた段数SWは段数ースイッチ112bであると判断されて1ページの段数を減少させる場合には、現在の段数nが1以下(n≤1)であるか否かを判断する(ステップS1104)。

【0084】ステップS1104において、現在の段数nが1以下(n≤1)であると判断された場合には、段数をそれ以上減少させないので、楽譜表示プログラムのルーチン(図7)へリターンする。

【0085】一方、ステップS1104において、現在の段数nが1以下(n≤1)でないと判断された場合には、nを1だけデクリメントし(ステップS1106)、ステップS1110へ進む。

【0086】また、ステップS1102において、オンされた段数SWは段数+スイッチ112aであると判断されて1ページの段数を増加させる場合には、nを1だけインクリメントし(ステップS1108)、ステップS1110へ進む。

【0087】ステップS1110においては、小節サイズの縦方向Vm=Vw/nの演算を行い小節サイズの縦

方向Vmを算出するとともに、フォントサイズ $f = \text{int}(Vm/10)$ （fは四捨五入）の演算を行いフォントサイズfを算出し、楽譜表示プログラムのルーチン（図7）へリターンする。

【0088】即ち、この段数SWにより1ページの段数の変更が指示されたので、それに応じて新たな小節サイズの縦方向Vmを算出するとともに、新たなフォントサイズfを算出している。これによって、新たな小節サイズの縦方向Vmおよび新たなフォントサイズfが設定され、実際の表示は表示ルーチン（図8）で更新される。

【0089】次に、図12に示すフローチャート参照しながら、Window変更処理ルーチンについて説明する。

【0090】このWindow変更処理ルーチンにおいては、まず、OSよりウィンドウサイズを取得し、音符表示領域サイズの横方向Hwと縦方向Vwとを設定する（ステップS1202）。

【0091】次に、小節サイズの横方向Hm=Hw/mの演算を行い小節サイズの横方向Hmを算出するとともに、小節サイズの縦方向Vm=Vw/nの演算を行い小節サイズの縦方向Vmを算出し、さらに、フォントサイズ $f = \text{int}(Vm/10)$ （fは四捨五入）の演算を行いフォントサイズfを算出し、楽譜表示プログラムのルーチン（図7）へリターンする（ステップS1204）。

【0092】即ち、このWindow変更処理ルーチンにおいては、変更されたウィンドウサイズに応じて新たな音符表示領域サイズの横方向Hwと縦方向Vwとを設定するとともに、新たな小節サイズの横方向Hmと縦方向Vmとを算出し、さらに新たなフォントサイズfを算出している。これによって、新たな音符表示領域サイズの横方向Hwおよび縦方向Vw、新たな小節サイズの横方向Hmおよび縦方向Vmならびに新たなフォントサイズfが設定され、実際の表示は表示ルーチン（図8）で更新される。

【0093】以上において詳細に説明したように、上記した第1の実施の形態によると、初期状態では図13(a)に示すように、ディスプレイ20の画面上には1段当たり4小節で2段に表示される。

【0094】ここで、例えば、音符の数が多い楽譜を表示するために1小節の幅を伸ばしたい場合には、ユーザーが小節-スイッチ110bを1回オンすることにより、図13(b)に示すように、1段当たりの小節数を1つ減らして、各小節の幅を伸ばすことができる。

【0095】また、図13(a)の表示態様において、例えば、多くの小節を一度に見たい場合には、ユーザーが段数+スイッチ112aを1回オンすることにより、図13(c)に示すように、1ページ当たりの表示段数を1段増やすことができる。なお、図13(c)の表示

態様においてはフォントサイズが小さくなるが、図13(d)に示すようにウィンドウサイズを大きくすれば、フォントサイズを大きくすることができる。

【0096】このように、上記した第1の実施の形態によれば、ユーザーの要望に応じて、適当な表示形態を適宜選択することができる。

【0097】なお、上記した第1の実施の形態においては、1段が10線間の幅に固定されているため、これを超える音高の音符は、段をはみ出して表示されることになる。こうした段をはみ出す表示を避けるためには、予め楽譜データの全ての音符の音高を読み出しておいて、その最高音と最低音とから1段の線間の数を決定するようすればよい。また、表示領域よりオクターブ以上離れた音符に対しては、オクターブ変更記号を附加して、表示領域内に収まるようにしてよい。

【0098】また、上記した第1の実施の形態においては、ユーザーの要望に応じて、小節数、段数を設定できるので、音符の数が多く1小節が長いような楽譜に合わせた表示形態をとることができる。

【0099】さらに、上記した第1の実施の形態においては、1段当たりの小節数ならびに段数の上限を設けなかつたが、上限を設定するようにしてよい。なお、上限を設定する場合には、音符表示領域の大きさに対応して上限を決定するようにしてよい。（2）第2の実施の形態：指定されたフォントサイズで楽譜表示を行う場合の処理、ウィンドウサイズの変更時には段の縦横比を固定してフォントサイズの変更を行う場合の処理、フォントサイズが変更されたとき段の縦横比の変更を行う場合の処理図14には、第2の実施の形態における楽譜表示プログラムによって表示されるウィンドウが示されている。このウィンドウ内には、マウス18によって操作可能な各種スイッチが表示されている。ユーザーはマウス18によってこれらのスイッチを操作することにより、所望の動作を指定できる。

【0100】また、これらのスイッチや音部記号、余白などの表示領域を除いた部分が音符表示領域100として設定されている。従って、音符表示領域100は、ウィンドウの形状変化に応じて、その形状が変化する。

【0101】さらに、リサイズボックス106が表示されており、ユーザーはこのリサイズボックス106にマウス18を合わせて、クリックしながら移動（ドラッグ）させることにより、このウィンドウの形状変更を指定できる。なお、ウィンドウの形状変化の情報はOSの処理によって検出され、楽譜表示プログラムに与えられる。

【0102】ここで、図14のウィンドウに示されるスイッチを説明すると、ページ+スイッチ108aは表示された楽譜のページを進めるスイッチであり、ページ-スイッチ108bは表示された楽譜のページを戻すスイッチであり、小節+スイッチ110aは1段当たりの表

示小節数を増加させるスイッチであり、小節ースイッチ110bは1段当たりの表示小節数を減少させるスイッチであり、フォント+スイッチ114aはフォントを1ドットづつ大きくするスイッチであり、フォントースイッチ114bはフォントを1ドットづつ小さくするスイッチである。

【0103】なお、ページ+スイッチ108aとページースイッチ108bとを総称してページスイッチ(ページSW)と称し、小節+スイッチ110aと小節ースイッチ110bとを総称して小節スイッチ(小節SW)と称し、フォント+スイッチ114aとフォントースイッチ114bとを総称してフォントスイッチ(フォントSW)と称する。

【0104】そして、この第2の実施の形態においては、上記した第1の実施の形態に用いた変数に加え、以下に示す変数が使用される。

【0105】即ち、

| | |
|-------------|------------------|
| ・段の縦横比 | R |
| ・ウィンドウ修正フラグ | Wch _g |
| ・段間サイズ | V _r |

の変数が、上記した第1の実施の形態に用いた変数に加えて用いられる。

【0106】以下、フローチャートを参照しながら説明するが、第1の実施の形態における各ルーチンのフローチャートの処理と同様な処理内容に関しては、第1の実施の形態と同一のステップ番号を付して示すことにより、その詳細な説明は省略する。

【0107】図15には、ユーザーの起動指示によりOSから起動される楽譜表示プログラムのルーチンのフローチャートが示されている。

【0108】即ち、第2の実施の形態における図15において示す楽譜表示プログラムのルーチンにおいては、ステップS1506において、変数の初期値として、m=4、n=2、St=1、Wch_g=0を設定し、H_m=H_w/m、V_m=V_w/n、f=int(V_m/10)を算出するとともに、段の縦横比R=V_m/H_wを算出する。ここで、fは、四捨五入された整数値である。

【0109】上記した各種の初期状態における設定を終了すると、この楽譜表示プログラムのルーチンのサブルーチンたる表示ルーチン(図16参照)を実行する(ステップS1508)。

【0110】また、ステップS1518においては、フォントSWがオンされたか否かを判断する。

【0111】そして、ステップS1518において、フォントSWがオンされたと判断された場合には、この楽譜表示プログラムのルーチンのサブルーチンたるフォントSW処理ルーチン(図17)を実行する(ステップS1520)。そして、フォントSW処理ルーチンを実行した後は、ステップS1508の表示ルーチンへ戻って処理を繰り返す。

【0112】なお、フォントSW処理ルーチンにおいては、図17を参照しながら後述するように、押されたフォントSWに応じてフォントサイズを大きくしたり、小さくしたりする処理を行う。フォントサイズは1段の縦に対応しているので、フォントサイズを変更すると、表示段数の再計算が行われるとともに、1段の縦横比も変更される。

【0113】一方、ステップS1518において、フォントSWはオンされていないと判断された場合には、ウインドウ(Window)の変化があったか否かを判断する(ステップS1522)。即ち、OSにウインドウサイズが変更されたか否かを問い合わせる。

【0114】そして、ステップS1522において、ウインドウサイズが変更されたと判断された場合には、この楽譜表示プログラムのルーチンのサブルーチンたるWindow変更処理ルーチン(図18)を実行する(ステップS1524)。そして、Window変更処理ルーチンを実行した後は、ステップS1508の表示ルーチンへ戻って処理を繰り返す。

【0115】なお、ページSW、小節SWあるいはフォントSWのいずれもオンではなく、ウインドウサイズも変更されていない場合には、ステップS710へ戻ってループして待機する。

【0116】次に、図16に示すフローチャート参照しながら、表示ルーチンについて説明する。なお、第1の実施の形態における図8に示す表示ルーチンのフローチャートの処理と同様な処理内容に関しては、図8と同一のステップ番号を付して示すことにより、その詳細な説明は省略する。

【0117】この表示ルーチンにおいては、まず、Wch_gが1であるか否かを判断する(ステップS1602)。

【0118】ステップS1602において、Wch_gが1であると判断されてウインドウサイズの修正が指示されている場合には、音符表示領域サイズH_w、V_wに対応したウインドウサイズに現在のウインドウサイズを変更するようにOSに指示し、さらに、Wch_gを0にしてウインドウ修正フラグをクリアし(ステップS1604)、ステップS1606の処理へ進む。

【0119】一方、ステップS1602において、Wch_gが1でない、即ち、Wch_gが0にクリアされている場合には、ステップS1606へ進む。

【0120】ステップS1606においては、段間サイズV_r=(V_w/n)-V_mの演算を行い、段間サイズV_rを算出する。

【0121】ステップS1606の処理を終了すると、段数のカウンタiを0にクリアし(ステップS802)、小節数のカウンタjを0にクリアする(ステップS804)。

【0122】次に、ウインドウ内の各種スイッチを表示

するとともに、縦が $i \times (V_m + V_r)$ ドットから $i \times (V_m + V_r) + V_m$ ドットまでの幅 V_m の表示領域に 5 線を表示し、音部記号、調性記号などを音符表示領域 100 外の 5 線上の所定の位置に表示する（ステップ S 1612）。これにより、各段間は V_r の間隔をもって、音符表示領域 100 に均等に割り付けられることになる。

【0123】次に、図 17 に示すフローチャート参照しながら、フォント SW 处理ルーチンについて説明する。

【0124】このフォント SW 处理ルーチンにおいては、まず、オンされたフォント SW はフォントースイッチ 114b であるか否かを判断する（ステップ S 1702）。ステップ S 1702 において、オンされたフォント SW はフォントースイッチ 114b でない、即ち、オンされたフォント SW はフォント+スイッチ 114a であると判断されてフォントを大きくする場合には、フォントサイズのテンポラリレジスタ $f' = f + 1$ の演算を行いフォントサイズのテンポラリレジスタ f を算出し、小節サイズの縦のテンポラリレジスタ $V_m' = f' \times 10$ の演算を行い小節サイズの縦のテンポラリレジスタ V_m' を算出し、段数のテンポラリレジスタ $n' = \text{int}(V_w / V_m')$ （ただし、 n' は切り捨て）の演算を行い段数のテンポラリレジスタ n' を算出する（ステップ S 1704）。

【0125】それから、 n' は 1 未満 ($n' < 1$) であるか否かを判断する（ステップ S 1706）。

【0126】ステップ S 1706 において、 n' は 1 未満 ($n' < 1$) であると判断された場合には、指定されたフォントサイズが大きすぎて、段の縦の幅が音符表示領域を超てしまっているので、フォント SW 处理ルーチンを終了して、楽譜表示プログラムのルーチン（図 15）ヘリターンする。

【0127】一方、ステップ S 1706 において、 n' は 1 未満 ($n' < 1$) でないと判断された場合には、 $f = f'$ 、 $V_m = V_m'$ 、 $n = n'$ として、各テンポラリレジスタを各変数にコピーし（ステップ S 1708）、ステップ S 1714 へ進む。一方、ステップ S 1702 において、オンされたフォント SW はフォントースイッチ 114b であると判断されてフォントを小さくする場合には、 f は 1 以下 ($f \leq 1$) であるか否かを判断する（ステップ S 1710）。

【0128】ステップ S 1710 において、 f は 1 以下 ($f \leq 1$) でありフォントサイズが最小になっている場合には、フォント SW 处理ルーチンを終了して、楽譜表示プログラムのルーチン（図 15）ヘリターンする。

【0129】一方、ステップ S 1710 において、 f は 1 以下 ($f \leq 1$) でないと判断された場合には、 $f = f - 1$ 、 $V_m = f \times 10$ 、 $n = \text{int}(V_w / V_m)$ （ただし、 n は切り捨て。）を算出し（ステップ S 1712）、ステップ S 1714 へ進む。

【0130】ステップ S 1714 においては、段の縦横比 $R = V_m / H_w$ を演算し、段の縦横比 R を算出する。これは、フォントサイズが変更されて段の高さが変更された場合に、段の縦横比をこれに応じて変更するものである。後述する Window 変更処理（図 18）においては、この変更された段の縦横比を使用して演算が行われる。そして、このステップ S 1714 の処理を終了すると、楽譜表示プログラムのルーチン（図 15）ヘリターンする。

【0131】つまり、このフォント SW 处理ルーチンにおいては、フォント SW によりフォントの変更が指示されたので、それに応じて小節の縦、即ち、段の縦を算出するとともに、新たな段の縦横比を算出しているものである。なお、小節の横は変更しない。

【0132】次に、図 18 に示すフローチャート参照しながら、Window 変更処理ルーチンについて説明する。

【0133】この Window 変更処理ルーチンにおいては、まず、OS よりウィンドウサイズを取得し、音符表示領域サイズの横方向 H_w と縦方向 V_w を設定する（ステップ S 1202）。

【0134】次に、小節サイズの横方向 $H_m = H_w / m$ の演算を行い小節サイズの横方向 H_m を算出し、小節サイズの縦方向 $V_m = H_w \times R$ の演算を行い小節サイズの縦方向 R を算出し、フォントサイズ $f = \text{int}(V_m / 10)$ （ f は四捨五入）の演算を行いフォントサイズ f を算出し、さらに、段数 $n = \text{int}(V_w / V_m)$ （ただし、 n は切り捨て。）の演算を行い段数 n を算出する（ステップ S 1804）。

【0135】ここでは、ウィンドウサイズが変更された場合は、段の縦横比を一定に保つようしている。

【0136】それから、 n は 1 未満 ($n < 1$) であるか否か判断する（ステップ S 1806）。

【0137】そして、ステップ S 1806 において、 n は 1 未満 ($n < 1$) でないと判断された場合には、楽譜表示プログラムのルーチン（図 15）ヘリターンする。

【0138】一方、ステップ S 1806 において、 n は 1 未満 ($n < 1$) であると判断された場合には、変更されたウィンドウサイズが小さすぎて、音符表示領域が段の縦の幅より狭くなってしまっているので、ウィンドウサイズを最低限必要な大きさに修正を行う（ステップ S 1808）。具体的には、 $V_w = V_m$ で音符表示領域の縦を段の縦に一致させ、 $n = 1$ とし、ウィンドウ修正フラグ W_{ch} に 1 をセットして、楽譜表示プログラムのルーチン（図 15）ヘリターンする。これにより、この Window 変更処理ルーチンを終了した後に、表示ルーチン（図 16）においてウィンドウサイズの修正が行われる。

【0139】即ち、Window 変更処理ルーチンにおいては、変更されたウィンドウサイズに応じて、段の縦

横比を一定にするように段の縦を算出し、これから新たな段部とフォントサイズとを算出している。

【0140】以上において詳細に説明したように、上記した第2の実施の形態によると、初期状態においては、上記した第1の実施の形態による図13(a)と同様に、1段当たり4小節で2段に表示される。

【0141】ここで、ユーザーが1小節の幅を変更したい場合には、第1の実施の形態の場合と同様に、小節SWの操作により1段当たりの小節数を増減させねばよい。

【0142】また、音符の数が多い楽譜などで、1段当たりの小節数は減らしたくないが、表示されている音符が詰まりすぎている場合は、フォントースイッチ114bでフォントサイズを小さくすれば。その場合には、図19(a)に示すように、1段の縦Vmが小さくなり、余った部分は段間サイズVrとして各段毎に均等に配分され、バランスのとれた割り付けがなされる。このとき、同時に段の縦横比R=Vm/Hwも減少される。

【0143】この後に、ウィンドウサイズを図19(b)に示すように縦に伸ばしたとすると、減少した段の縦横比Rはそのまま段数の計算が行われ、例えば、3段が表示できるようになる。

【0144】このように、ウィンドウサイズの変更に対して、段の形状バランスを保つようにすることができる。

【0145】なお、この第2の実施の形態においては、フォントを指定するのにフォントサイズfドットを指定したが、指定するのは段の高さの値にしておいて、フォントサイズは計算によって求めてよい。

【0146】また、上記した第2の実施の形態においては、段の形状比率を保つことで、表示領域の形状やサイズに関わらず、段のイメージを保ったままで、拡大/縮小を行うことができる。(3) 第3の実施の形態: 小節の縦横比を固定で表示を行う処理

図20には、第3の実施の形態における楽譜表示プログラムによって表示されるウィンドウが示されている。このウィンドウ内には、マウス18によって操作可能な各種スイッチが表示されている。ユーザーはマウス18によってこれらのスイッチを操作することにより、所望の動作を指定できる。

【0147】また、これらのスイッチや音部記号、余白などの表示領域を除いた部分が音符表示領域100として設定されている。従って、音符表示領域100は、ウィンドウの形状変化に応じて、その形状が変化する。

【0148】さらに、リサイズボックス106が表示されており、ユーザーはこのリサイズボックス106にマウス18を合わせて、クリックしながら移動(ドラッグ)させることにより、このウィンドウの形状変更を指定できる。なお、ウィンドウの形状変化の情報はOSの処理によって検出され、楽譜表示プログラムに与えられ

る。

【0149】ここで、図20のウィンドウに示されるスイッチを説明すると、ページ+スイッチ108aは表示された楽譜のページを進めるスイッチであり、ページ-スイッチ108bは表示された楽譜のページを戻すスイッチであり、小節+スイッチ110aは1段当たりの表示小節数を増加させるスイッチであり、小節-スイッチ110bは1段当たりの表示小節数を減少させるスイッチであり、段数+スイッチ112aは表示段数を増加させるスイッチであり、段数-スイッチ112bは表示段数を減少させるスイッチであり、フォント+スイッチ114aはフォントを1ドットづつ大きくするスイッチであり、フォント-スイッチ114bはフォントを1ドットづつ小さくするスイッチである。

【0150】さらに、図20のウィンドウには、優先モードスイッチ(優先モードSW)116が示されている。この優先モードスイッチ116は、ウィンドウサイズの変更のときに、どの状態を優先して保存するかを選択するためのスイッチであり、1段あたり小節数、段数、フォントサイズの三者択一である。モードMの値としては、小節(M=1)、段数(M=2)、フォントサイズ(M=3)の値をとるものとする。

【0151】なお、ページ+スイッチ108aとページ-スイッチ108bとを総称してページスイッチ(ページSW)と称し、小節+スイッチ110aと小節-スイッチ110bとを総称して小節スイッチ(小節SW)と称し、段数+スイッチ112aと段数-スイッチ112bとを総称して段数スイッチ(段数SW)と称する。そして、この第3の実施の形態においては、上記した第1の実施の形態ならびに第2の実施の形態に用いた変数に加え、以下に示す変数が使用される。

【0152】即ち、

| | |
|-------------|------------------|
| ・小節の縦横比 | r |
| ・ウィンドウ修正フラグ | Wch _g |
| ・段間サイズ | Vr |
| ・モード | M |

の変数が、上記した第1の実施の形態に用いた変数に加えて用いられる。

【0153】以下、フローチャートを参照しながら説明するが、第1の実施の形態ならびに第2の実施の形態における各ルーチンのフローチャートの処理と同様な処理内容に関しては、第1の実施の形態ならびに第2の実施の形態におけるステップ番号と同一のステップ番号をして示すことにより、その詳細な説明は省略する。

【0154】図21には、ユーザーの起動指示によりOSから起動される楽譜表示プログラムのルーチンのフローチャートが示されている。

【0155】即ち、第3の実施の形態における図21に示す楽譜表示プログラムのルーチンにおいては、ステップS2106において、変数の初期値として、m=4、

$r = 0, 33, S t = 1$ を設定し、 $Hm = Hw/m, Vm = r \times Hm, n = \text{int}(Vw/Vm), f = \text{int}(Vm/10)$ を算出する。ここで、 n は切り捨てられた整数値であり、 f は四捨五入された整数値である。

【0156】上記した各種の初期状態における設定を終了すると、この楽譜表示プログラムのルーチンのサブルーチンたる表示ルーチンを実行する（ステップS2108）。

【0157】また、ステップS1518においては、フォントSWがオンされたか否かを判断する。

【0158】そして、ステップS1518において、フォントSWがオンされたと判断された場合には、この楽譜表示プログラムのルーチンのサブルーチンたるフォントSW処理ルーチン（図24）を実行する（ステップS2122）。そして、フォントSW処理ルーチンを実行した後は、ステップS2108の表示ルーチンへ戻って処理を繰り返す。

【0159】なお、フォントSW処理ルーチンにおいては、図24を参照しながら後述するように、押されたフォントSWに応じてフォントサイズを大きくしたり、小さくしたりする処理を行う。フォントサイズは1段の縦に対応しているので、フォントサイズを変更すると、表示段数の再計算が行われるとともに、1段当たりの小節数の再計算も行われる。

【0160】一方、ステップS1518において、フォントSWはオンされていないと判断された場合には、ウィンドウ（Window）の変化があったか否かを判断する（ステップS2124）。即ち、OSにウィンドウサイズが変更されたか否かを問い合わせる。

【0161】そして、ステップS2124において、ウィンドウサイズが変更されたと判断された場合には、この楽譜表示プログラムのルーチンのサブルーチンたるWindow変更処理ルーチン（図25）を実行する（ステップS2126）。そして、Window変更処理ルーチンを実行した後は、ステップS2108の表示ルーチンへ戻って処理を繰り返す。

【0162】一方、ステップS2124において、ウィンドウサイズが変更されていないと判断された場合には、優先モードスイッチ116の変更があったか否かを判断する（ステップS2128）。

【0163】ステップS2128において、優先モードスイッチ116の変更があったと判断された場合には、モードMに対応する値を読み込む（ステップS2130）。なお、ページSW、小節SW、段数SWあるいはフォントSWのいずれもオンではなく、ウィンドウサイズも変更されておらず、さらには優先モードスイッチ116も変更されていない場合には、ステップS710へ戻ってループして待機する。

【0164】なお、ステップS2108の表示ルーチンは、第2の実施の形態における表示ルーチン（図16）

とほぼ同じ処理内容であるので、その図示を省略した。なお、第3の実施の形態においては、ステップS1606において、 $Hm = Hw/m$ の処理が追加される。

【0165】これは、小節の縦横比を一定に保とうとした場合、左右に余白または不足が生じることがあるが、その場合に、各小節を均等に引き伸ばしたりあるいは縮めたりして、音符表示領域内にちょうど収まるようにするものである。

【0166】次に、図22に示すフローチャート参照しながら、小節SW処理ルーチンについて説明する。

【0167】この小節SW処理ルーチンにおいては、まず、オンされた小節SWは小節+スイッチ110aであるか否かを判断する（ステップS1002）。

【0168】ステップS1002において、オンされた小節SWは小節+スイッチ110aでない、即ち、オンされた小節SWは小節-スイッチ110bであると判断されて1段当たりの小節数を減少させる場合には、現在の小節数mが1以下（ $m \leq 1$ ）であるか否かを判断する（ステップS1004）。

【0169】ステップS1004において、現在の小節数mが1以下（ $m \leq 1$ ）であると判断された場合には、小節数をそれ以上減少させられないので、楽譜表示プログラムのルーチン（図21）へリターンする。

【0170】一方、ステップS1004において、現在の小節数mが1以下（ $m \leq 1$ ）でないと判断された場合には、小節数のテンポラリレジスタ $m' = m - 1$ 、小節サイズの横方向のテンポラリレジスタ $Hm' = Hw/m'$ 、小節サイズの縦方向のテンポラリレジスタ $Vm' = r \times Hm'$ および段数のテンポラリレジスタ $n' = \text{int}(Vw/Vm')$ （ただし、 n' は切り捨て。）の演算を行い、小節数のテンポラリレジスタ m' 、小節サイズの横方向のテンポラリレジスタ Hm' 、小節サイズの縦方向のテンポラリレジスタ Vm' および段数のテンポラリレジスタ n' を算出する（ステップS2206）。

【0171】それから、 n' は1未満（ $n' < 1$ ）であるか否かを判断する（ステップS2208）。

【0172】ステップS2208において、 n' は1未満（ $n' < 1$ ）であると判断された場合には、段の縦の幅が音符表示領域を越えてしまっているので、小節SWルーチンの処理を中断して楽譜表示プログラムのルーチン（図21）へリターンする。

【0173】一方、ステップS2208において、 n' は1未満（ $n' < 1$ ）でないと判断された場合には、 $m = m'$ 、 $Hm = Hm'$ 、 $Vm =Vm'$ および $n = n'$ としてテンポラリレジスタを各変数にコピーするとともに、 $f = \text{int}(Vm/10)$ （ただし、 f は四捨五入。）を算出して、楽譜表示プログラムのルーチン（図21）へリターンする（ステップS2210）。

【0174】また、ステップS1002において、オン

された小節SWは小節+スイッチ110aであると判断されて1段当たりの小節数を増加させる場合には、 $m = m + 1$ 、 $Hm = Hw/m$ 、 $Vm = r \times Hm$ 、 $n = int(Vw/Vm)$ （ただし、nは切り捨て。）、 $f = int(Vm/10)$ （ただし、fは四捨五入。）の演算を行い、m、Hm、Vm、nおよびfを算出し（ステップS2212）、楽譜表示プログラムのルーチン（図21）へリターンする。

【0175】即ち、この小節SW処理ルーチンにおいては、小節SWにより1段当たりの小節数の変更が指示されたので、それに応じて新たな小節サイズの縦方向Hmを算出し、小節の縦横比が一定になるように小節の縦方向Vmを算出し、段数、フォントサイズを算出しており、実際の表示は表示ルーチン（図8）で更新される。次に、図23に示すフローチャート参照しながら、段数SW処理ルーチンについて説明する。

【0176】この段数SW処理ルーチンにおいては、まず、オンされた段数SWは段数+スイッチ112aであるか否かを判断する（ステップS1102）。

【0177】ステップS1102において、オンされた段数SWは段数+スイッチ112aでない、即ち、オンされた段数SWは段数-スイッチ112bであると判断されて1ページの段数を減少させる場合には、現在の段数nが1以下（ $n \leq 1$ ）であるか否かを判断する（ステップS1104）。

【0178】ステップS1104において、現在の段数nが1以下（ $n \leq 1$ ）であると判断された場合には、段数をそれ以上減少させないので、楽譜表示プログラムのルーチン（図21）へリターンする。

【0179】一方、ステップS1104において、現在の段数nが1以下（ $n \leq 1$ ）でないと判断された場合には、段数のテンポラリレジスタ $n' = n - 1$ 、小節サイズの縦方向のテンポラリレジスタ $Vm' = Vw/n$ 、小節サイズの横方向のテンポラリレジスタ $Hm' = Vm/r$ および1段当たりの小節数のテンポラリレジスタ $m' = int(Hw/Hm)$ （ただし、m'は四捨五入。）の演算を行い、段数のテンポラリレジスタ n' 、小節サイズの縦方向のテンポラリレジスタ Vm' 、小節サイズの横方向のテンポラリレジスタ Hm' および1段当たりの小節数のテンポラリレジスタ m' を算出する（ステップS2306）。

【0180】それから、m'は1未満（ $m' < 1$ ）であるか否かを判断する（ステップS2308）。

【0181】ステップS2308において、m'は1未満（ $m' < 1$ ）であると判断された場合には、指定された段数を表示しようとすると、音符表示領域サイズの横方向Hwが小節サイズの横方向Hmに比べて小さく表示ができないので、この場合は段数SW処理ルーチンを中断して楽譜表示プログラムのルーチン（図21）へリターンする。

【0182】一方、ステップS2308において、m'は1未満（ $m' < 1$ ）でないと判断された場合には、 $n = n'$ 、 $Vm = Vm'$ 、 $Hm = Hm'$ および $m = m'$ としてテンポラリレジスタを各変数にコピーし（ステップS2310）、ステップS2314へ進む。

【0183】また、ステップS1102において、オンされた段数SWは段数+スイッチ112aであると判断されて1ページの段数を増加させる場合には、 $n = n + 1$ 、 $Vm = Vw/n$ 、 $Hm = Vm/r$ および $m = int(Hw/Hm)$ （ただし、m'は四捨五入。）の演算を行い、n、Vm、Hmおよびmを算出する（ステップS2312）し、ステップS2314へ進む。

【0184】ステップS2314においては、フォントサイズ $f = int(Vm/10)$ （ただし、fは四捨五入。）を算出して、楽譜表示プログラムのルーチン（図21）へリターンする。（ステップS2210）。

【0185】即ち、この段数SW処理ルーチンにおいては、段数SWにより1ページの段数の変更が指示されたので、それに応じて新たな小節サイズの縦方向Vmを算出し、小節の縦横比が一定になるように小節サイズの横方向Hmを算出し、新たな1段当たりの小節数mを算出するとともに、新たなフォントサイズfを算出しており、実際の表示は表示ルーチン（図2108）で更新される。

【0186】次に、図24に示すフローチャート参照しながら、フォントSW処理ルーチンについて説明する。

【0187】このフォントSW処理ルーチンにおいては、まず、オンされたフォントSWはフォント-スイッチ114bであるか否かを判断する（ステップS1702）。ステップS1702において、オンされたフォントSWはフォント-スイッチ114bでない、即ち、オンされたフォントSWはフォント+スイッチ114aであると判断されてフォントを大きくする場合には、フォントサイズのテンポラリレジスタ $f' = f + 1$ 、小節サイズの縦方向のテンポラリレジスタ $Vm' = f' \times 10$ 、小節サイズの横方向のテンポラリレジスタ $Hm' = Vm/r$ 、1段当たりの小節数のテンポラリレジスタ $m' = int(Hw/Hm)$ （ただし、m'は四捨五入。）および段数のテンポラリレジスタ $n' = int(Vw/Vm')$ （ただし、n'は切り捨て。）の演算を行い、フォントサイズのテンポラリレジスタ f' 、小節サイズの縦方向のテンポラリレジスタ Vm' 、小節サイズの横方向のテンポラリレジスタ Hm' 、1段当たりの小節数のテンポラリレジスタ m' および段数のテンポラリレジスタ n' を算出する（ステップS2404）。

【0188】それから、m'は1未満（ $m' < 1$ ）あるいはn'は1未満（ $n' < 1$ ）であるか否かを判断する（ステップS2406）。

【0189】ステップS2406において、m'は1未満（ $m' < 1$ ）あるいはn'は1未満（ $n' < 1$ ）であ

ると判断された場合には、指定されたフォントサイズによって小節サイズを変更しようとすると、音符表示領域の縦あるいは横をはみ出してしまい表示ができなくなるので、フォントSW処理ルーチンを終了して、楽譜表示プログラムのルーチン(図21)へリターンする。

【0190】一方、ステップS2406において、 m' は1未満($m' < 1$)あるいは n' は1未満($n' < 1$)でないと判断された場合には、 $f = f'$ 、 $Vm = Vm'$ 、 $Hm = Hm'$ 、 $m = m'$ および $n = n'$ として、各テンポラリレジスタを各変数にコピーし(ステップS2408)、楽譜表示プログラムのルーチン(図21)へリターンする。

【0191】また、ステップS1702において、オニされたフォントSWはフォントースイッチ114bであると判断されてフォントを小さくする場合には、 f は1以下($f \leq 1$)であるか否か判断する(ステップS1710)。

【0192】ステップS1710において、 f は1以下($f \leq 1$)でありフォントサイズが最小になっている場合には、フォントSW処理ルーチンを終了して、楽譜表示プログラムのルーチン(図21)へリターンする。

【0193】一方、ステップS1710において、 f は1以下($f \leq 1$)でないと判断された場合には、 $f = f - 1$ 、 $Vm = f \times 10$ 、 $m = \text{int}(Hw/Hm)$ (ただし、 m は四捨五入。)ならびに $n = \text{int}(Vw/Vm)$ (ただし、 n は切り捨て。)を算出し(ステップS2412)、楽譜表示プログラムのルーチン(図21)へリターンする。

【0194】つまり、このフォントSW処理ルーチンにおいては、フォントSWによりフォントサイズの変更が指示されたので、それに応じて小節サイズの縦方向Vmを算出し、小節の縦横比が一定になるように小節サイズの横方向Hmを算出し、新たな1段当たりの小節数m算出するとともに、新たな段数nを算出しているものである。

【0195】次に、図25に示すフローチャート参照しながら、Window変更処理ルーチンについて説明する。

【0196】このWindow変更処理ルーチンにおいては、まず、OSよりウィンドウサイズを取得し、音符表示領域サイズの横方向Hwと縦方向Vwとを設定する(ステップS1202)。

【0197】次に、M=1に設定されていて優先モードが小節になっているか否かについて判断する(ステップS2504)。

【0198】ステップS2504において、M=1に設定されていないと判断された場合には、M=2に設定されていて優先モードが段数になっているか否かについて判断する(ステップS2506)。

【0199】ステップS2506において、M=2に設

定されていないと判断された場合、即ち、M=1でもなくM=2でもなく、優先モードがM=3のフォントになっている場合には、現在のフォントサイズfに基づいて、 $Vm = f \times 10$ 、 $Hm = Vm/r$ 、 $m = \text{int}(Hw/Hm)$ (ただし、 m は四捨五入。)および $n = \text{int}(Vw/Vm)$ (n は、四捨五入。)を算出する(ステップS2508)。

【0200】それから、 m は1未満($m < 1$)であるか否かを判断する(ステップS2510)。

【0201】ステップS2510において、 m は1未満($m < 1$)である場合には、変更された音符表示領域サイズの横方向Hwが1小節よりも小さくなってしまっているので、ウィンドウサイズを最低限必要な大きさに修正する。即ち、 $m = 1$ 、 $Hw = Hm$ および $Wchg = 1$ に設定する(ステップS2512)。

【0202】これにより、このWindow変更処理ルーチンを終了した後に、表示ルーチン(ステップS2108)においてウィンドウサイズの修正が行われる。

【0203】ステップS2512の処理を終了した場合、あるいはステップS2510において m は1未満($m < 1$)でないと判断された場合には、 n は1未満($n < 1$)であるか否かを判断する(ステップS2514)。

【0204】ステップS2514において、 n は1未満($n < 1$)であると判断された場合には、変更された音符表示領域サイズの横方向Vwが小節の縦よりも小さくなってしまっているので、ウィンドウサイズを最低限必要な大きさに修正する。即ち、 $n = 1$ 、 $Vw = Vm$ および $Wchg = 1$ に設定する(ステップS2516)。これにより、このWindow変更処理ルーチンを終了した後に、表示ルーチン(ステップS2108)においてウィンドウサイズの修正が行われる。

【0205】ステップS2516の処理を終了した場合、あるいはステップS2514において n は1未満($n < 1$)でないと判断された場合には、楽譜表示プログラムのルーチン(図21)へリターンする。

【0206】また、ステップS2506において、M=2に設定されていると判断された場合、即ち、優先モードがM=2の段数になっている場合には、現在の段数nに基づいて、 $Vm = Vw/n$ および $Hm = Vm/r$ 、 $m = \text{int}(Hw/Hm)$ (ただし、 m は四捨五入。)を算出する(ステップS2518)。

【0207】それから、 m は1未満($m < 1$)であるか否かを判断する(ステップS2520)。

【0208】ステップS2520において、 m は1未満($m < 1$)である場合には、変更された音符表示領域サイズの横方向Hwが1小節よりも小さくなってしまっているので、ウィンドウサイズを最低限必要な大きさに修正する。即ち、 $m = 1$ 、 $Hw = Hm$ および $Wchg = 1$ に設定する(ステップS2522)。

【0209】これにより、このWindow変更処理ルーチンを終了した後に、表示ルーチン（ステップS2108）においてウィンドウサイズの修正が行われる。

【0210】ステップS2522の処理を終了した場合、あるいはステップS2520においてmは1未満（ $m < 1$ ）でないと判断された場合には、フォントサイズ $f = \text{int}(Vm / 10)$ （ただし、fは四捨五入。）を算出し（ステップS2524）、楽譜表示プログラムのルーチン（図21）へリターンする。

【0211】また、ステップS2504において、M=1に設定されていると判断された場合、即ち、優先モードがM=1の小節になっている場合には、現在の1段当たりの小節数mに基づいて、 $Hm = Hw / m$ 、 $Vm = r \times Hm$ および $n = \text{int}(Vw / Vm)$ （ただし、nは切り捨て。）を算出する（ステップS2526）。

【0212】それから、nは1未満（ $n < 1$ ）であるか否かを判断する（ステップS2528）。

【0213】ステップS2528において、nは1未満（ $n < 1$ ）であると判断された場合には、変更された音符表示領域サイズの縦方向Vwが小節の縦よりも小さくなってしまっているので、ウィンドウサイズを最低限必要な大きさに修正する。即ち、 $n = 1$ 、 $Vw = Vm$ および $Wch = 1$ に設定する（ステップS2530）。これにより、このWindow変更処理ルーチンを終了した後に、表示ルーチン（ステップS2108）においてウィンドウサイズの修正が行われる。

【0214】ステップS2530の処理を終了した場合、あるいはステップS2528においてnは1未満（ $n < 1$ ）でないと判断された場合には、フォントサイズ $f = \text{int}(Vm / 10)$ （ただし、fは四捨五入。）を算出し（ステップS2532）、楽譜表示プログラムのルーチン（図21）へリターンする。

【0215】即ち、このWindow変更処理ルーチンにおいては、変更されたウィンドウサイズと指定された優先モードに応じて、小節の縦横比を一定にすることなく、小節の大きさ、1段当たりの小節数、段数などを算出している。これによって、新たな音符表示領域サイズの横方向Hwおよび縦方向Vw、新たな小節サイズの横方向Hmおよび縦方向Vmならびに新たなフォントサイズfが設定され、実際の表示は表示ルーチン（ステップS2108）で更新される。

【0216】以上において詳細に説明したように、上記した第3の実施の形態によると、初期状態では図26（a）に示すように、ディスプレイ20の画面上には1段当たり4小節で2段に表示される。

【0217】ここで、例えば、ユーザーが小節+スイッチ110aを1回オンすることにより、1段当たりの小節数を1小節増やした場合には、図26（b）に示すように、小節の縦横比が一定なので、小節の縦が短くなり、段間Vrとして配分されて段が割り付けられ、フォ

ントも小節の縦に応じて小さくなる。

【0218】また、例えば、ユーザーが段数+スイッチ112aを1回オンすることにより、段数を1段増やした場合には、図26（c）に示すように、小節の縦横比が一定なので、小節の横が短くなり、1段当たりの小節数が増加し、フォントも小節の縦に応じて小さくなる。

【0219】また、例えば、ユーザーがフォント+スイッチ114aを1回オンすることにより、フォントを大きくした場合には、図26（d）に示すように、小節の縦横比が一定なので、小節の縦横とともに大きくなり、小節数、段数ともに減少する。また、優先モードが小節になっているときに、ウィンドウサイズを大きくすると、図26（e）に示すように、小節数が固定されているので、小節の横が長くなり、また、小節の縦横比が固定なので、小節の縦も長くなる。従って、フォントも大きくなる。なお、縦の余白が生じた場合は、段間Vrとして配分されて段が割り付けられる。

【0220】また、優先モードが段数になっているときに、ウィンドウサイズを大きくすると、図26（f）に示すように、段数が固定されているので、小節の縦が長くなり、また、小節の縦横比が固定なので、小節の横も長くなる。従って、フォントも大きくなる。

【0221】また、優先モードがフォントになっているときに、ウィンドウサイズを大きくすると、図26（g）に示すように、小節の縦横が変わらずに、小節数、段数が増加する。なお、縦の余白が生じた場合は、段間Vrとして配分されて段が割り付けられる。

【0222】なお、上記した第3の実施の形態においては、小節の縦横比は一定であったが、小節当たりの音符の密度によって変更してもよい。このようにすれば、小節内の音符が多い楽譜の場合には、小節の形状を横長にすることで、ユーザーにとってより見易くすることができる。

【0223】また、拍子に応じて小節の縦横比を換えてよい。そのようにすると、拍子が多い楽曲の楽譜は小節の形状が横長になり、ユーザーにとって拍子の感覚が捉えやすくなる。

【0224】さらにまた、1段当たりの小節数の代わりに、小節の横のドット数に対応する値を指定するようにしてもよい。

【0225】また、上記した第3の実施の形態においては、音符の数がそれほど多くない通常の曲の場合には、小節の形状比率を保つことで、表示領域の形状やサイズに関わらず、バランスのとれた楽譜表示を提供することができる。

【0226】

【発明の効果】本発明は、以上説明したように構成されているので、画面の表示領域がどのような形状やサイズであっても楽譜が欠けて表示されることがなく、ユーザーが視認し易い形態で楽譜を表示することができるとい

う優れた効果を奏する。

【図面の簡単な説明】

【図1】従来の楽譜表示方法の問題点を説明するための説明図である。

【図2】従来の楽譜表示方法の問題点を説明するための説明図である。

【図3】本発明による楽譜表示方法を実施するためのパソコン用コンピュータなどの汎用のコンピュータの全体構成を示すブロック構成図である。

【図4】本発明の楽譜表示方法による楽譜表示の処理の原理を説明するための説明図である。

【図5】記号のデータのフォーマットを示す説明図である。

【図6】第1の実施の形態における楽譜表示プログラムによって表示されるウィンドウを示す説明図である。

【図7】第1の実施の形態における楽譜表示プログラムのルーチンを示すフローチャートである。

【図8】第1の実施の形態における表示ルーチンを示すフローチャートである。

【図9】第1の実施の形態におけるページSWルーチンを示すフローチャートである。

【図10】第1の実施の形態における小節SWルーチンを示すフローチャートである。

【図11】第1の実施の形態における段数SWルーチンを示すフローチャートである。

【図12】第1の実施の形態におけるWindow変更処理ルーチンを示すフローチャートである。

【図13】第1の実施の形態における楽譜の表示形態を示す説明図である。

【図14】第2の実施の形態における楽譜表示プログラムによって表示されるウィンドウを示す説明図である。

【図15】第2の実施の形態における楽譜表示プログラムのルーチンを示すフローチャートである。

【図16】第2の実施の形態における表示ルーチンを示すフローチャートである。

【図17】第2の実施の形態におけるFont SWルーチンを示すフローチャートである。

【図18】第2の実施の形態におけるWindow変更処理ルーチンを示すフローチャートである。

【図19】第2の実施の形態における楽譜の表示形態を示す説明図である。

【図20】第3の実施の形態における楽譜表示プログラムによって表示されるWindowを示す説明図である。

【図21】第3の実施の形態における楽譜表示プログラムのルーチンを示すフローチャートである。

【図22】第3の実施の形態における小節SWルーチンを示すフローチャートである。

【図23】第3の実施の形態における段数SWルーチンを示すフローチャートである。

【図24】第3の実施の形態におけるFont SW処理ルーチンを示すフローチャートである。

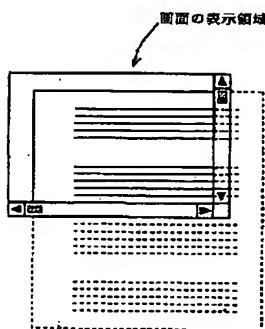
【図25】第3の実施の形態におけるWindow変更処理ルーチンを示すフローチャートである。

【図26】第3の実施の形態における楽譜の表示形態を示す説明図である。

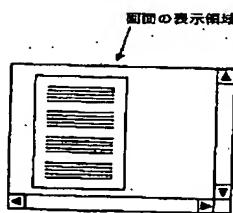
【符号の説明】

| | |
|-----|--------|
| 10 | CPU |
| 12 | RAM |
| 14 | 外部記憶装置 |
| 16 | キーボード |
| 18 | マウス |
| 20 | ディスプレイ |
| 100 | 音符表示領域 |
| 102 | 加線 |
| 104 | 線間 |

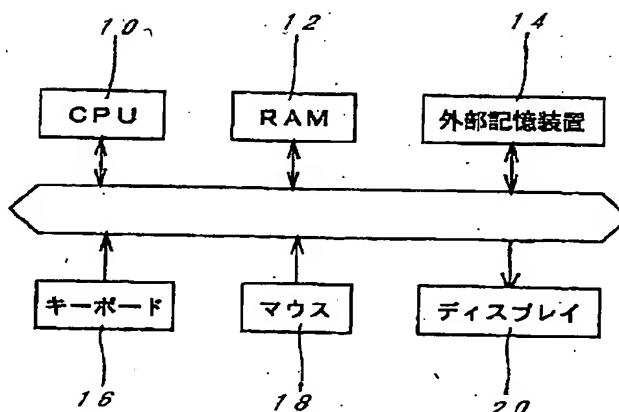
【図1】



【図2】



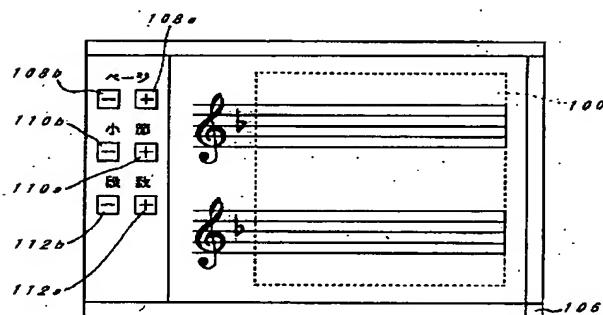
【図3】



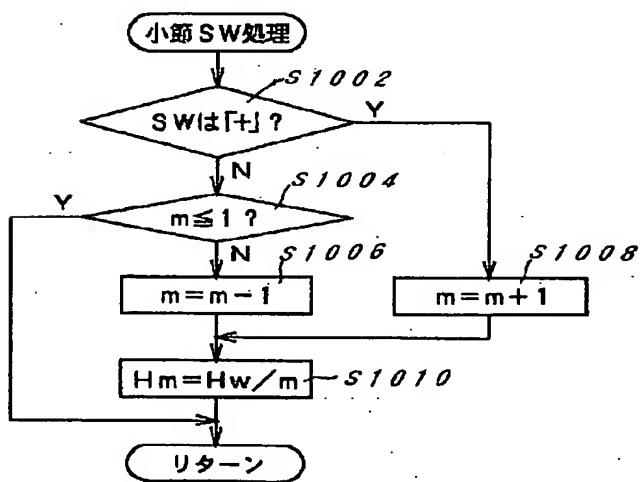
【図4】



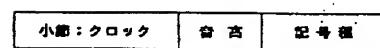
【図6】



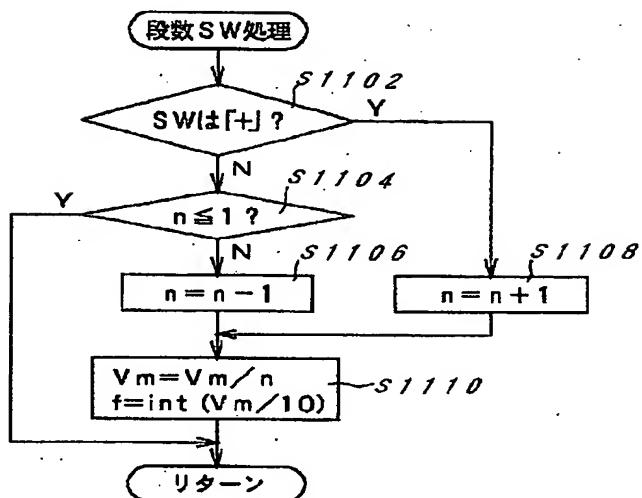
【図10】



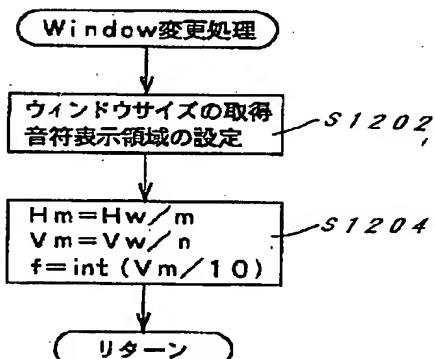
【図5】



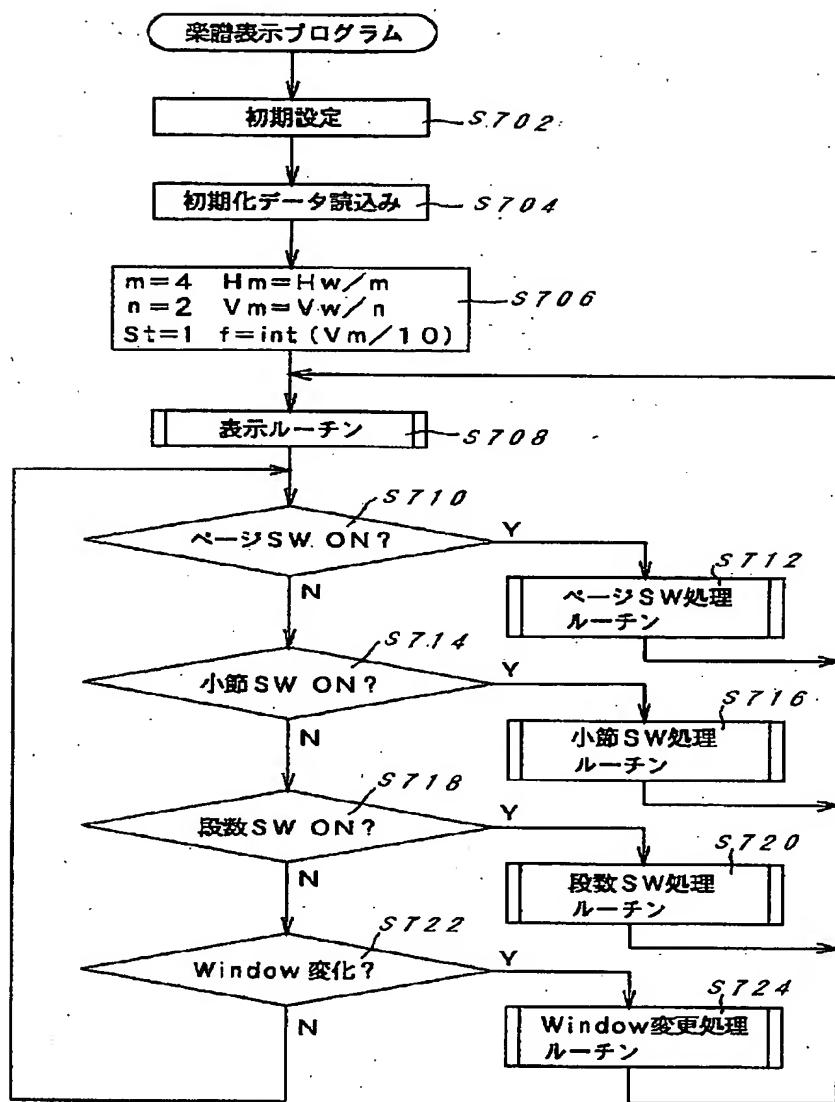
【図11】



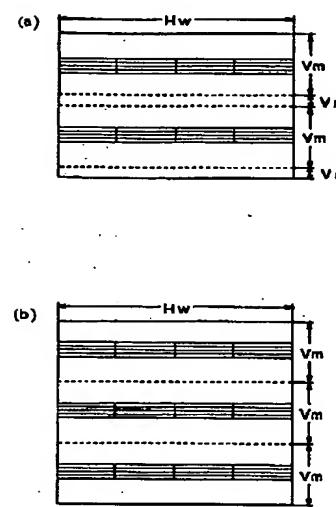
【図12】



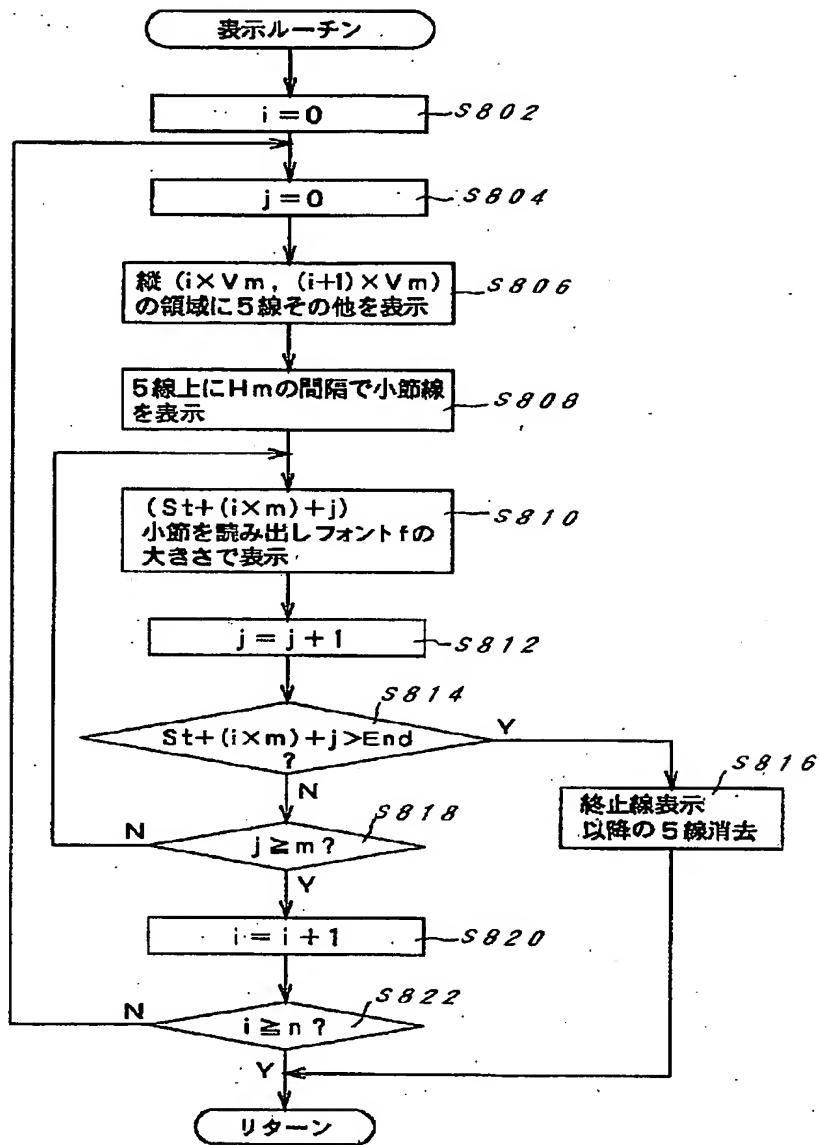
【図7】



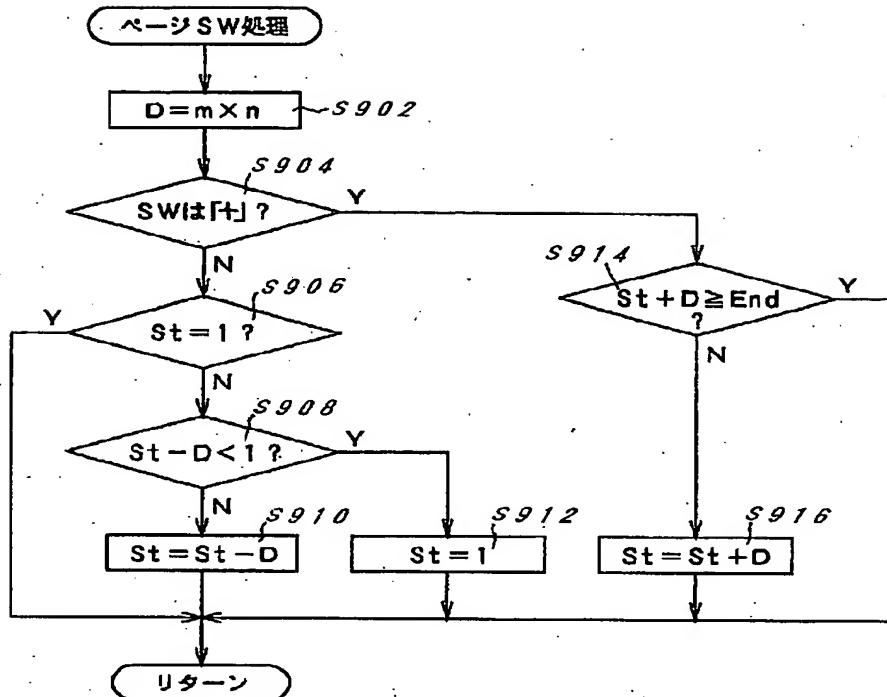
【図19】



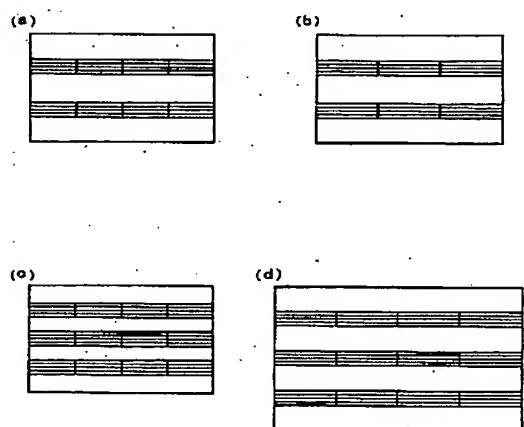
【図8】



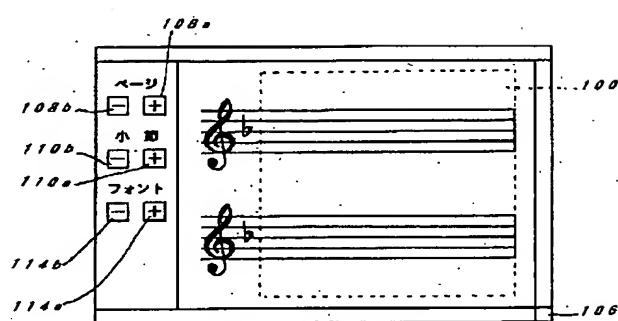
【図9】



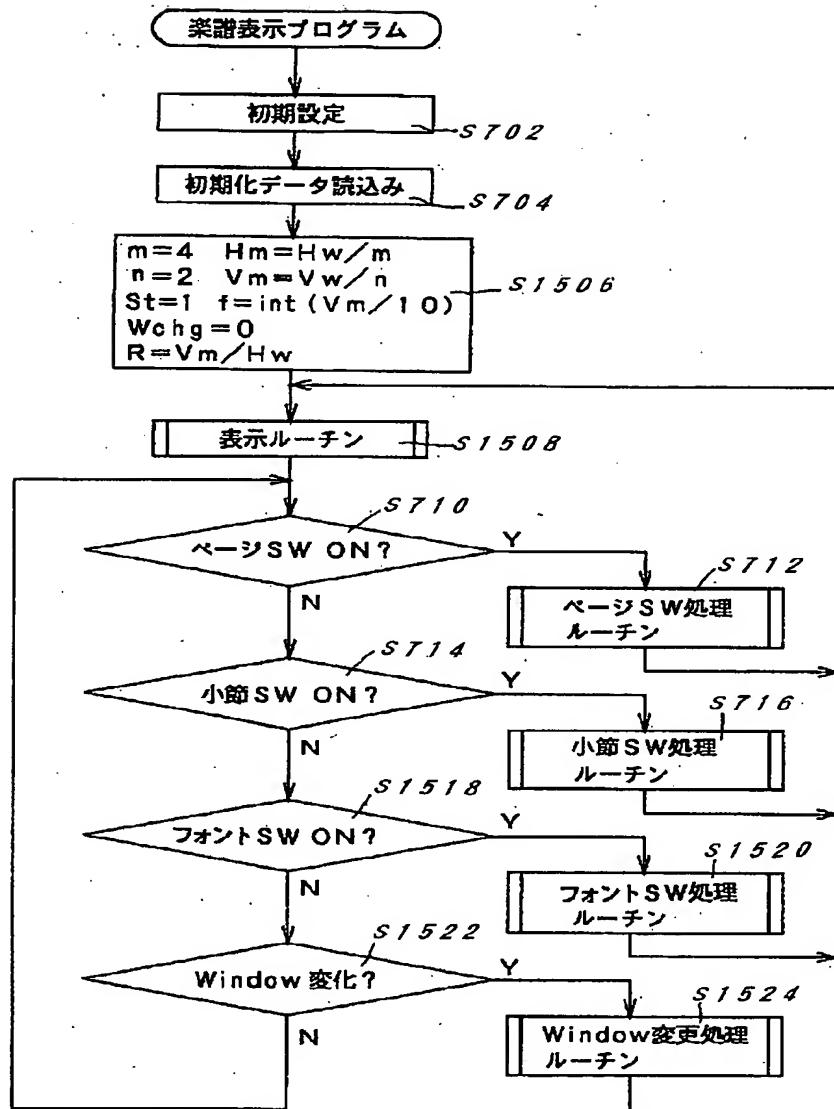
【図13】



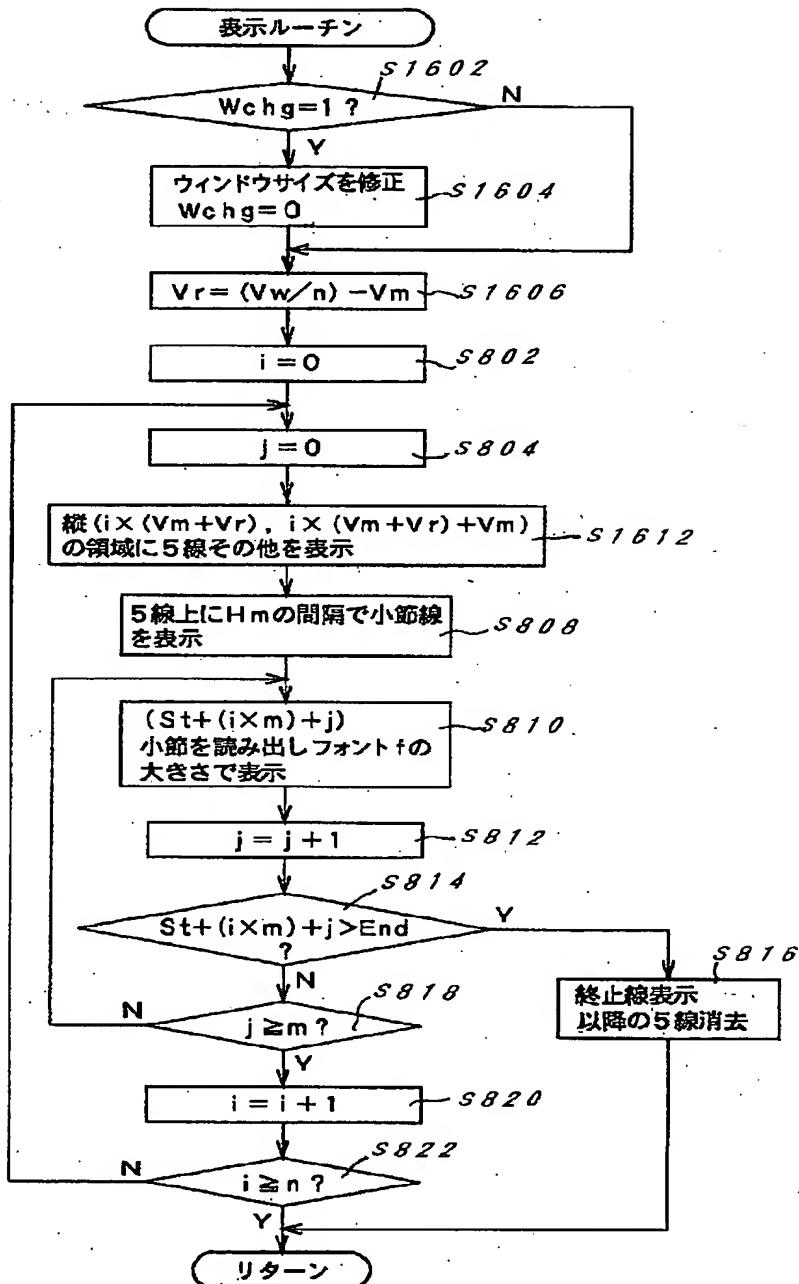
【図14】



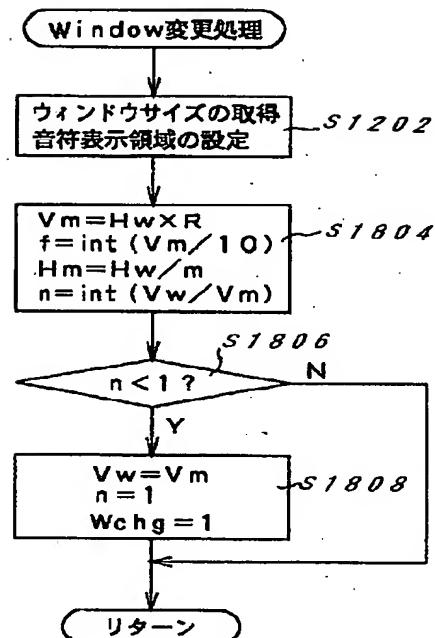
【図15】



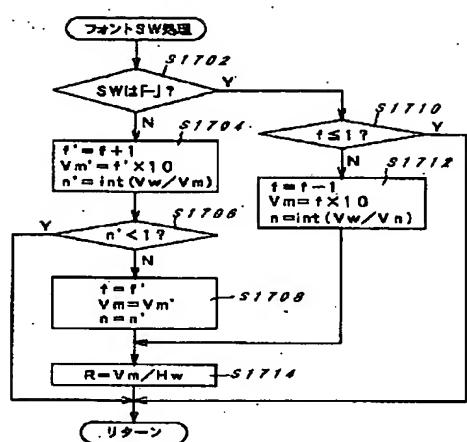
【図16】



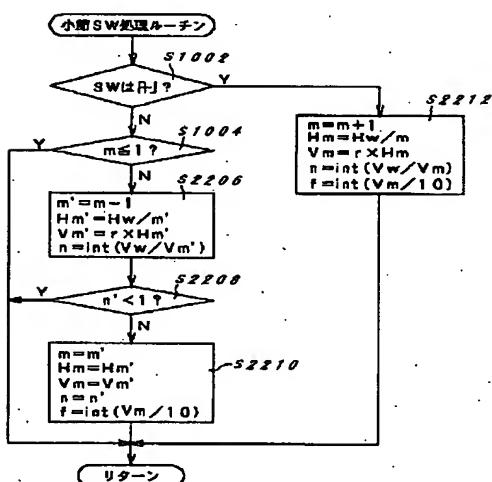
【図18】



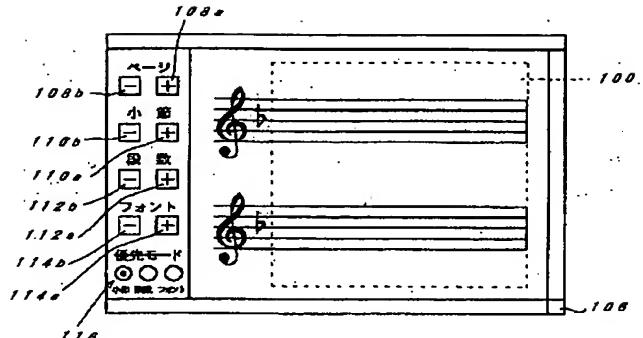
【图17】



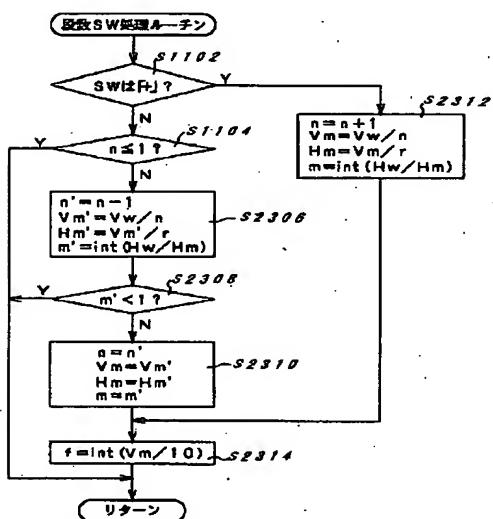
【图22】



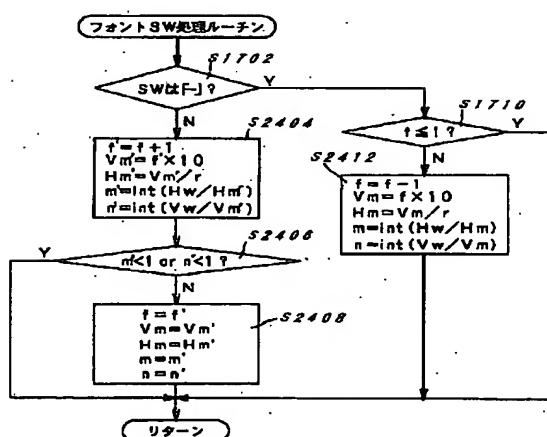
【四20】



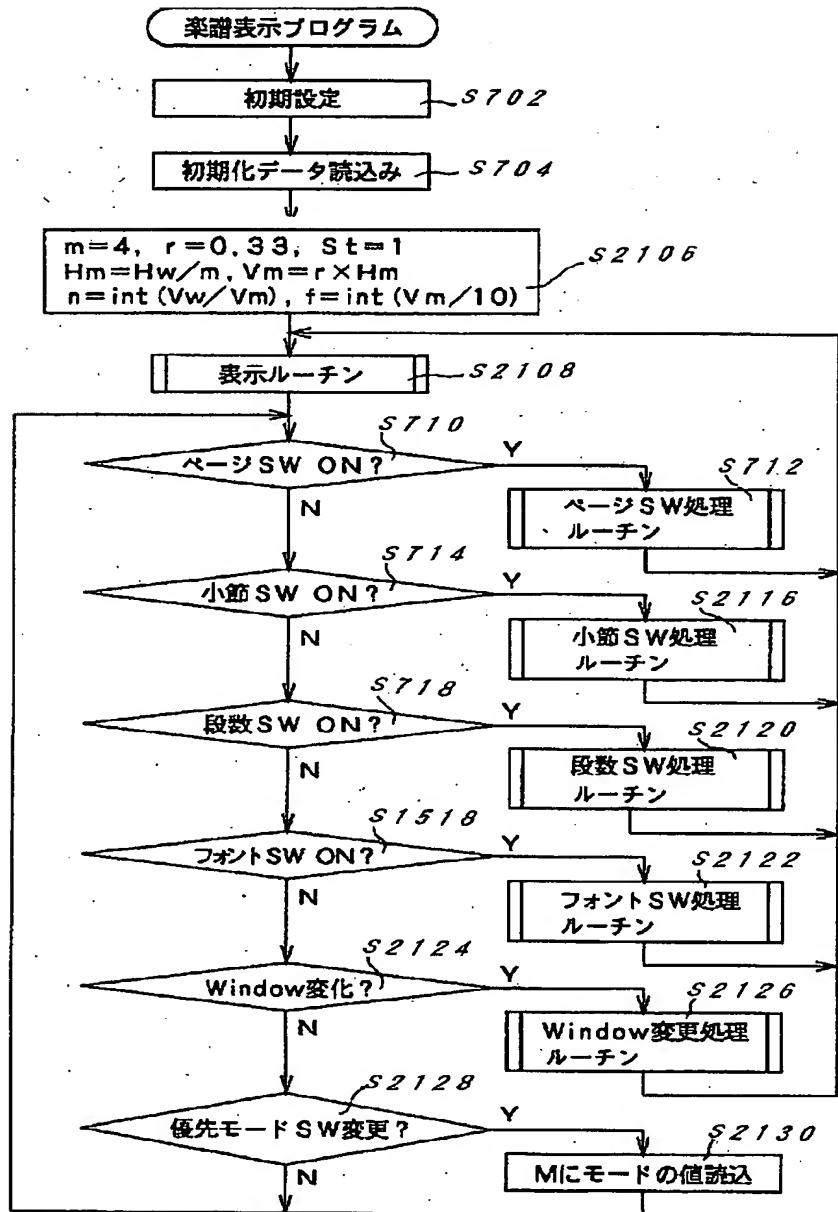
【図23】



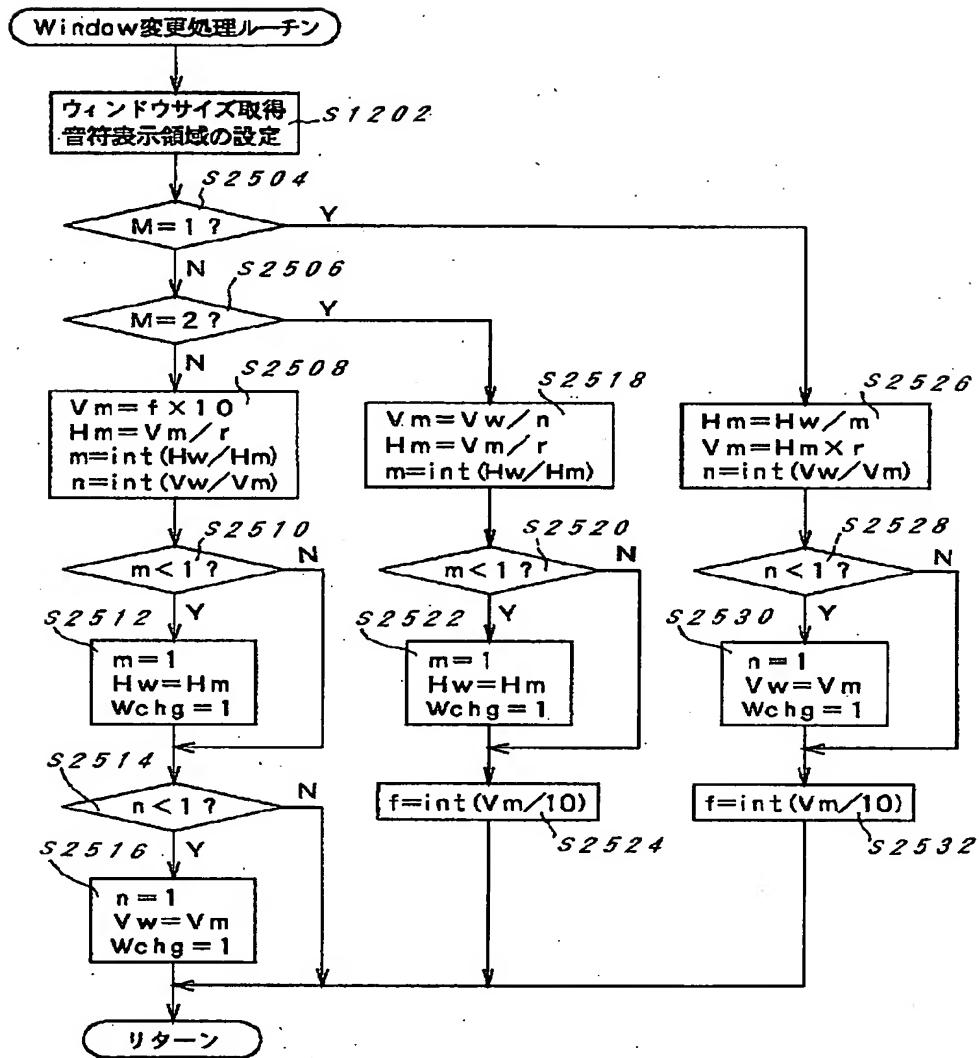
【図24】



【図21】



【図25】



【図26】

